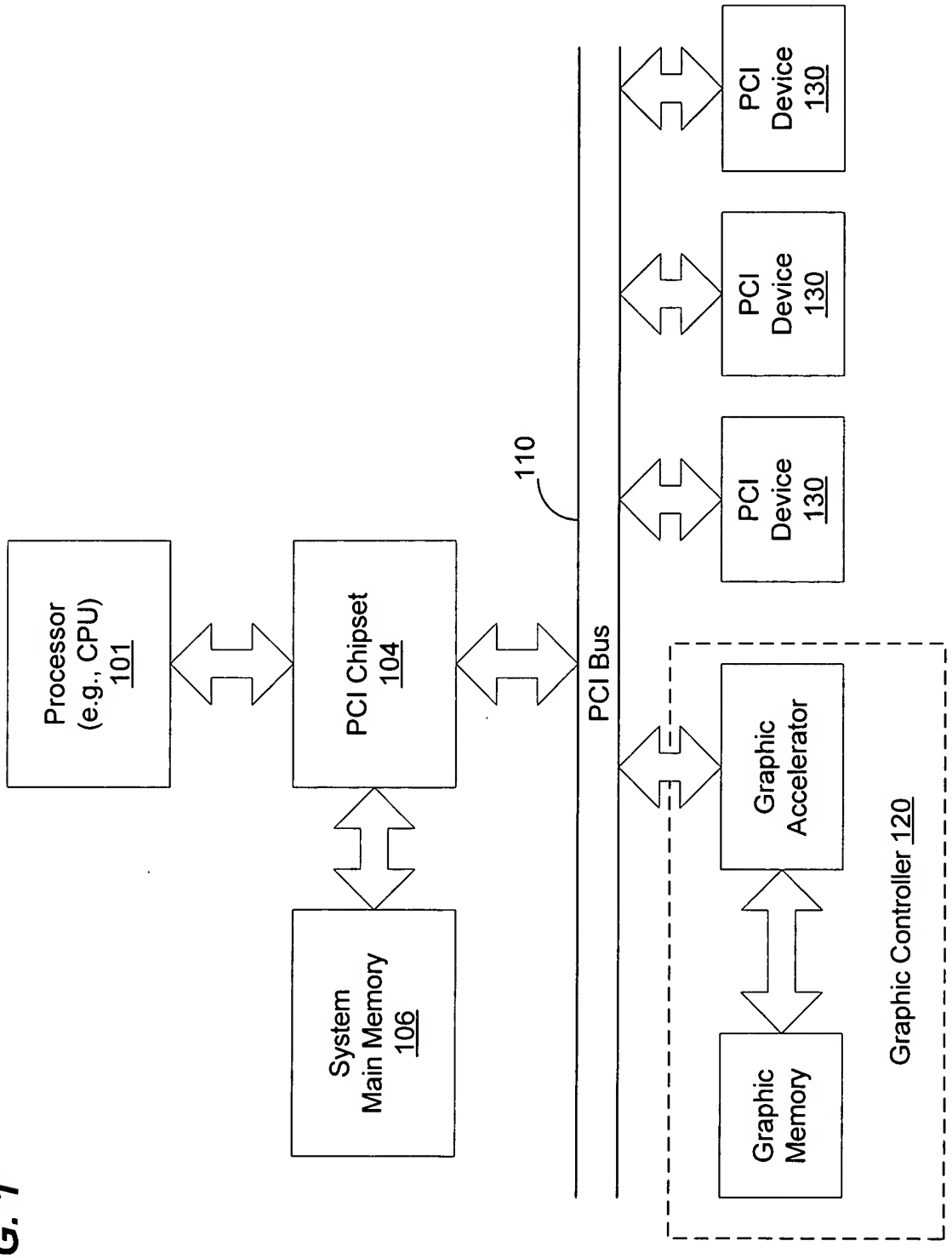
Processor (e.g., CPU) 101
PCI Chipset 104
System Main Memory 106
110
PCI Bus
PCI Device 130
PCI Device 130
PCI Device 130
Graphic Accelerator
Graphic Memory
Graphic Controller 120

FIG. 1

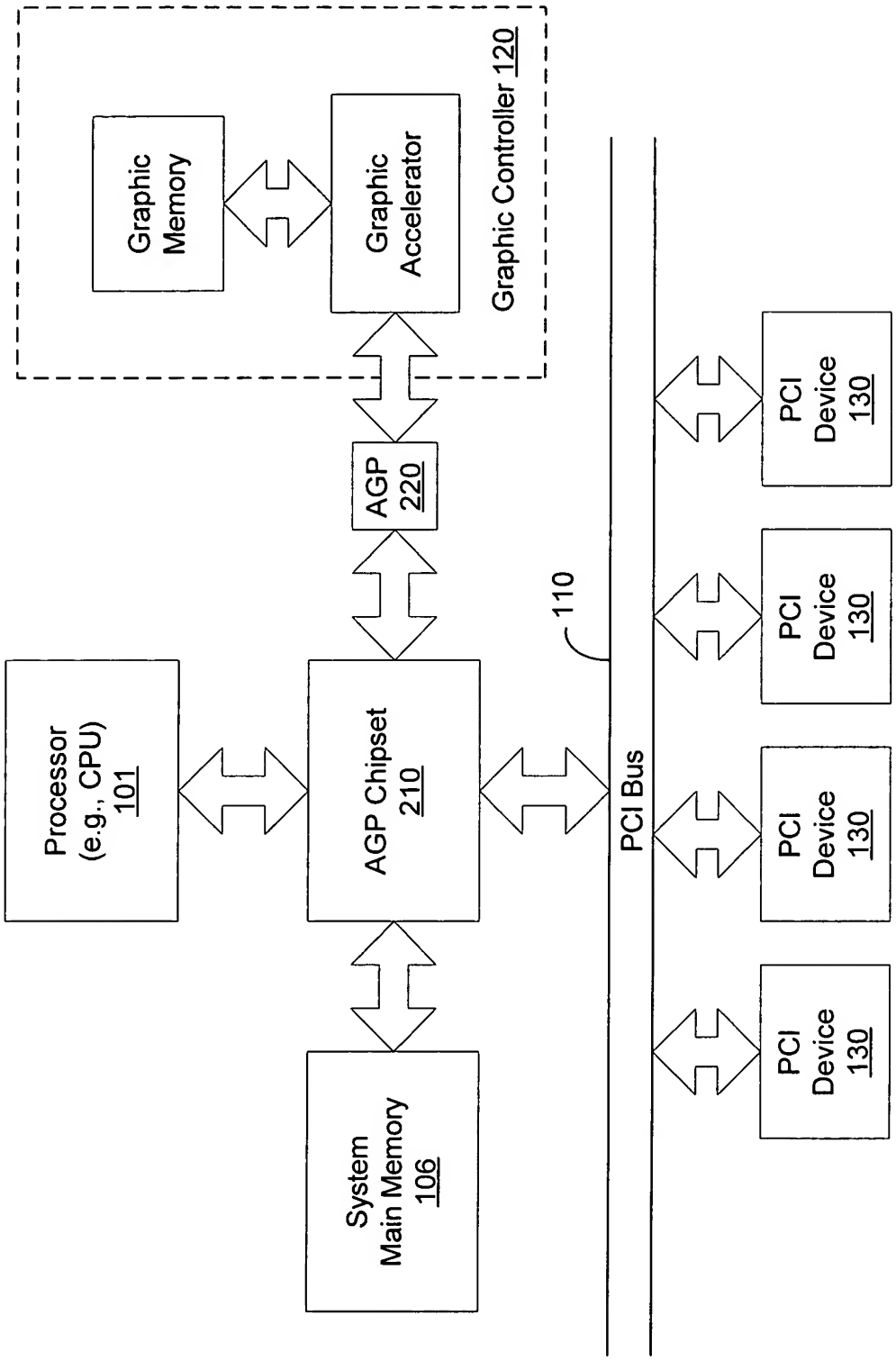
Graphic Controller 120
Graphic Memory
Graphic Accelerator
AGP 220
Processor (e.g., CPU) 101
AGP Chipset 210
System Main Memory 106
110
PCI Bus
PCI Device 130
PCI Device 130
PCI Device 130
PCI Device 130

FIG. 2

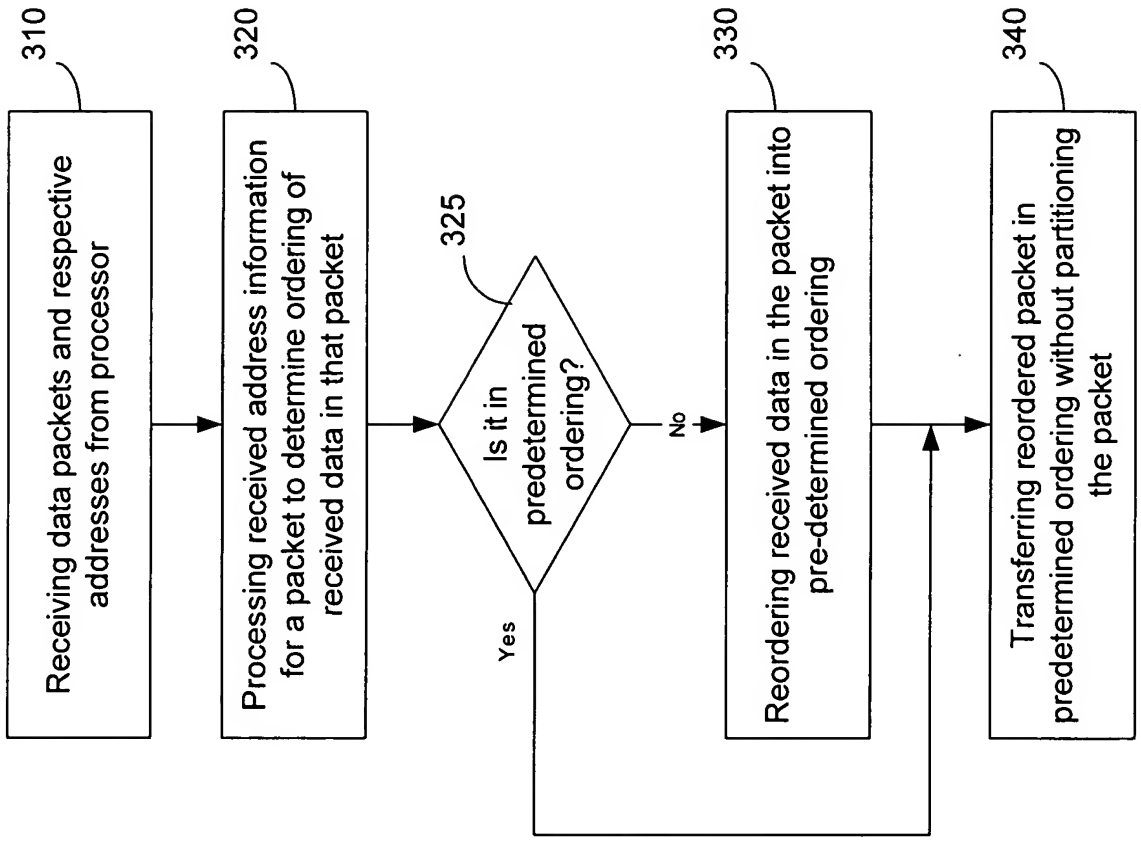
310
Receiving data packets and respective addresses from processor
320
Processing received address information for a packet to determine ordering of received data in that packet
325
Is it in predetermined ordering?
Yes
No
330
Reordering received data in the packet into pre-determined ordering
340
Transferring reordered packet in predetermined ordering without partitioning the packet
300

FIG. 3

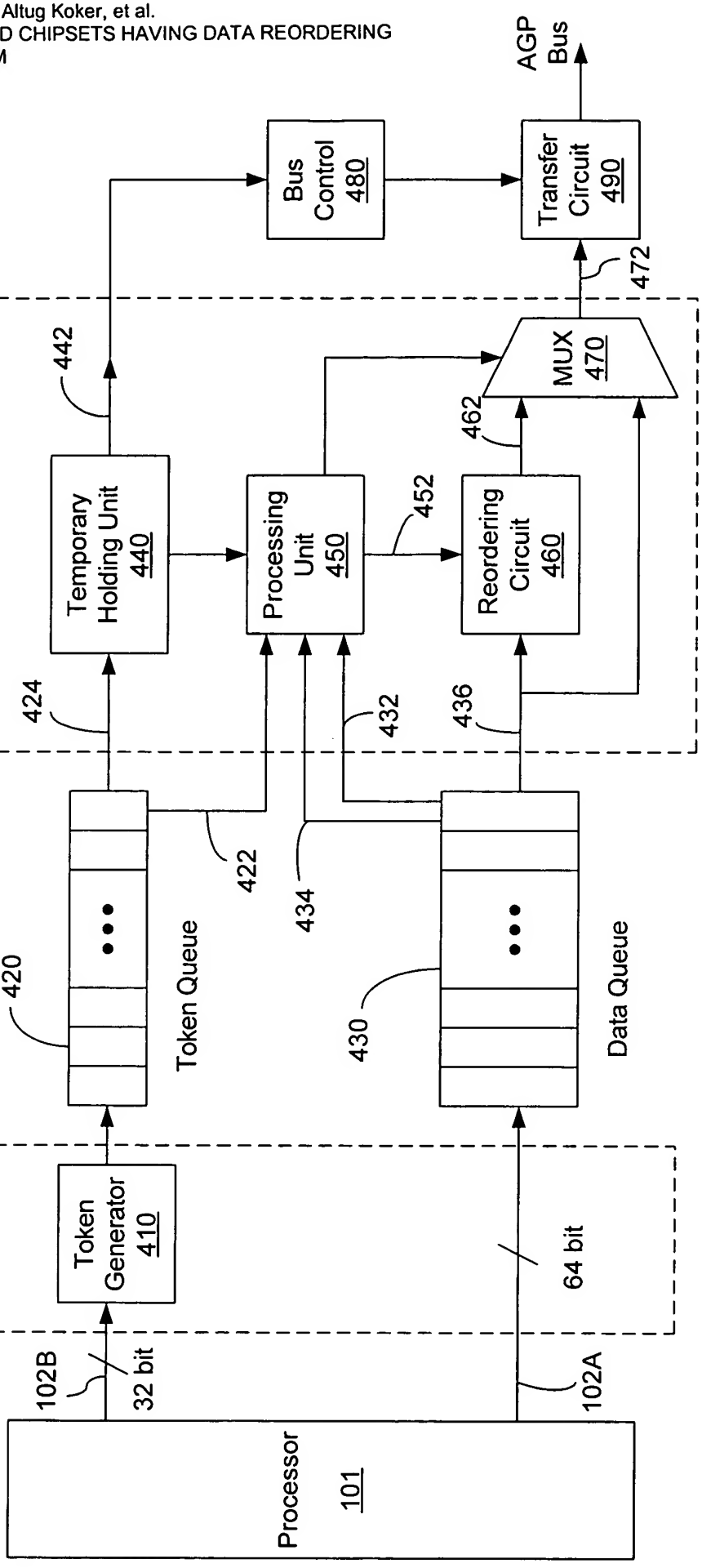
400
Processor
101
102B
32 bit
102A
401
Token Generator
410
64 bit
Token Queue
420
Data Queue
430
422
434
402
424
432
436
Temporary Holding Unit
440
Processing Unit
450
452
Reordering Circuit
460
462
MUX
470
442
472
Bus Control
480
Transfer Circuit
490
AGP Bus

FIG. 4

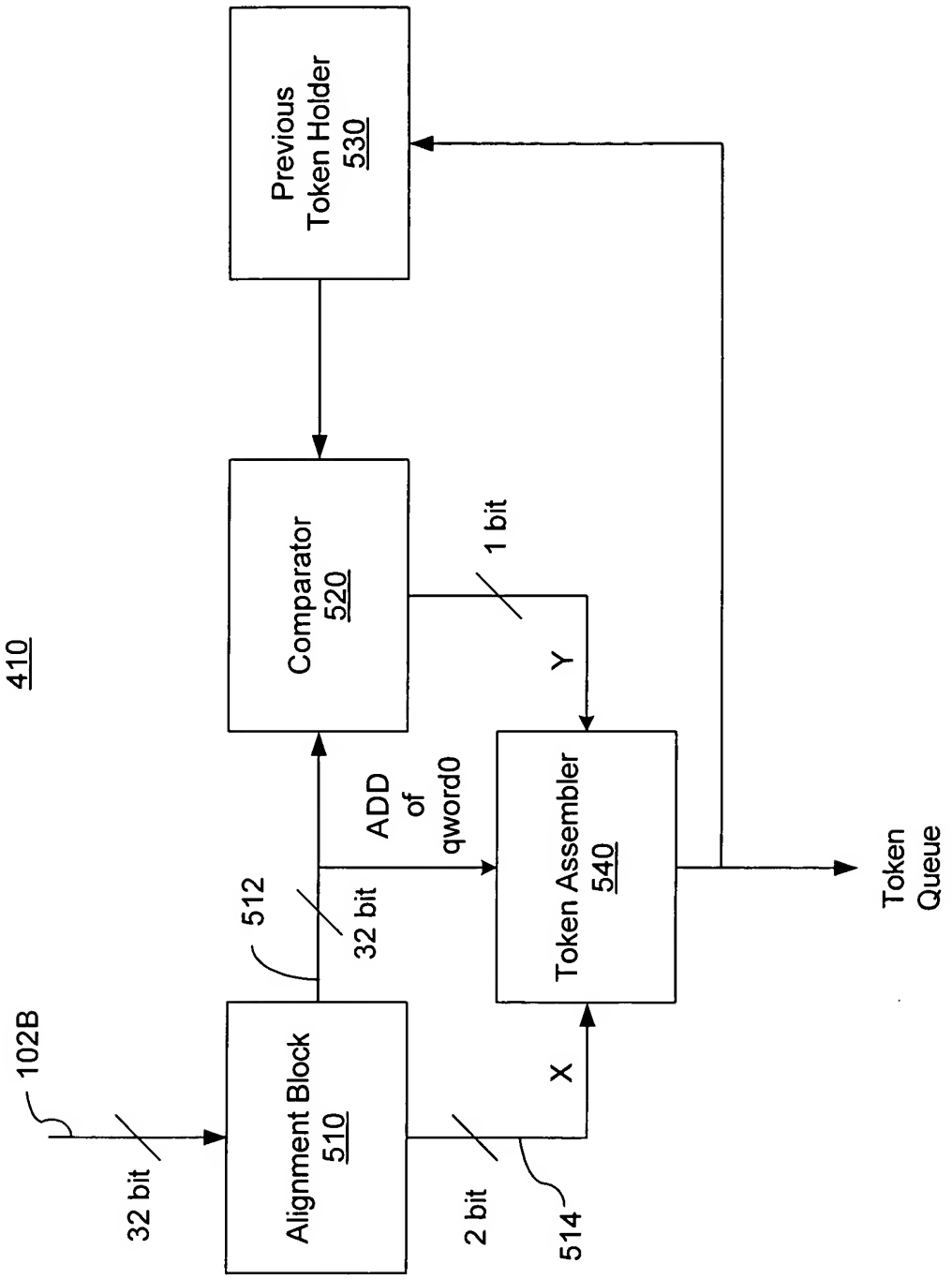
410
Previous Token Holder 530
Comparator 520
1 bit
Y
ADD of qword0
Token Assembler 540
Token Queue
512
32 bit
102B
Alignment Block 510
X
32 bit
2 bit
514

FIG. 5

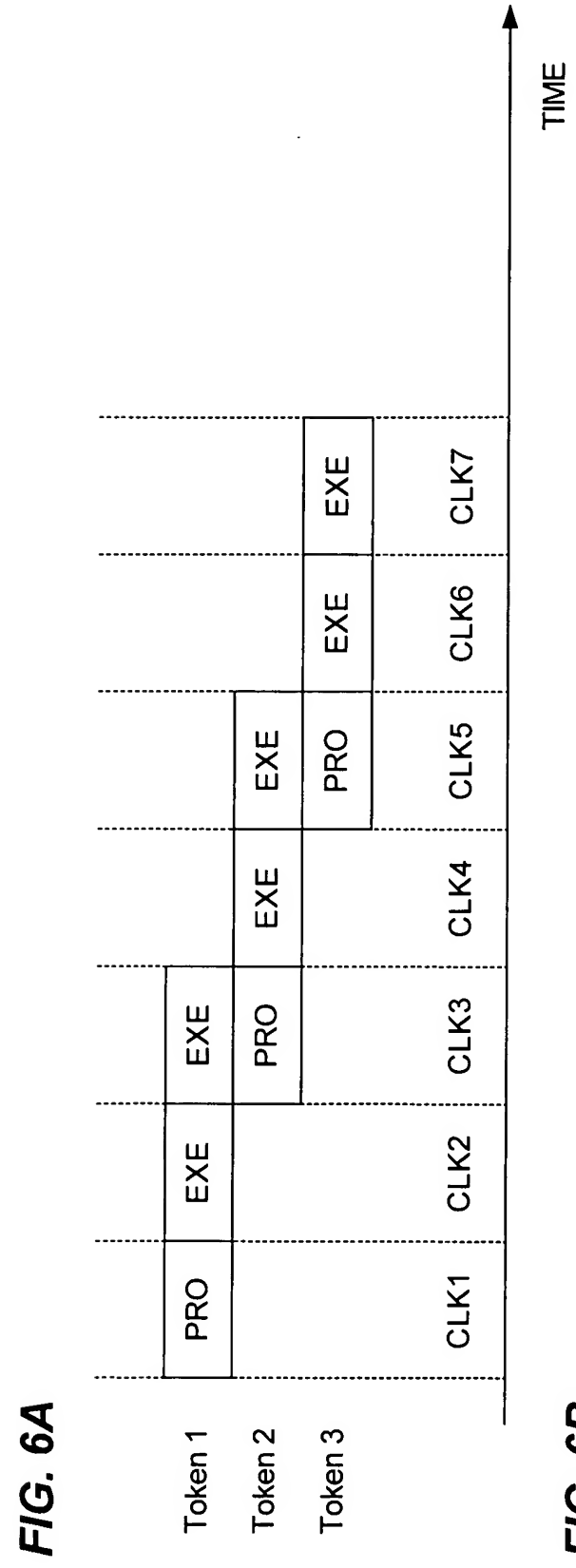
Token 1
Token 2
Token 3
PRO
EXE
EXE
EXE
EXE
EXE
EXE
PRO
EXE
EXE
EXE
PRO
EXE
EXE
PRO
EXE
CLK1
CLK2
CLK3
CLK4
CLK5
CLK6
CLK7
TIME

FIG. 6A

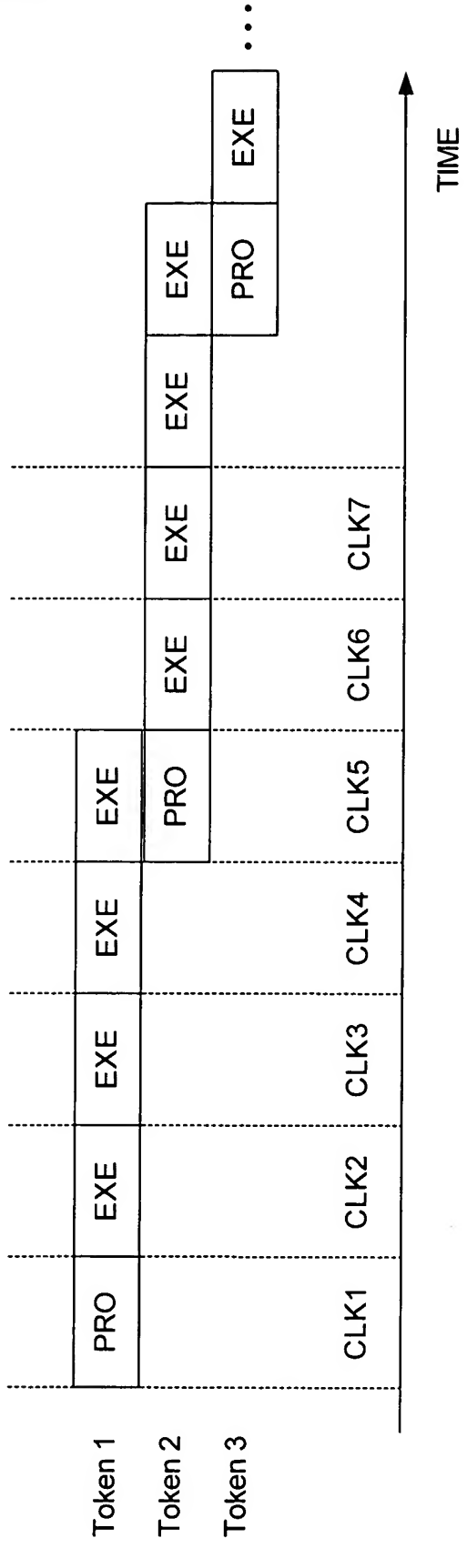
Token 1
Token 2
Token 3
PRO
EXE
EXE
EXE
EXE
EXE
EXE
PRO
EXE
EXE
EXE
PRO
EXE
EXE
PRO
EXE
CLK1
CLK2
CLK3
CLK4
CLK5
CLK6
CLK7
TIME

FIG. 6B

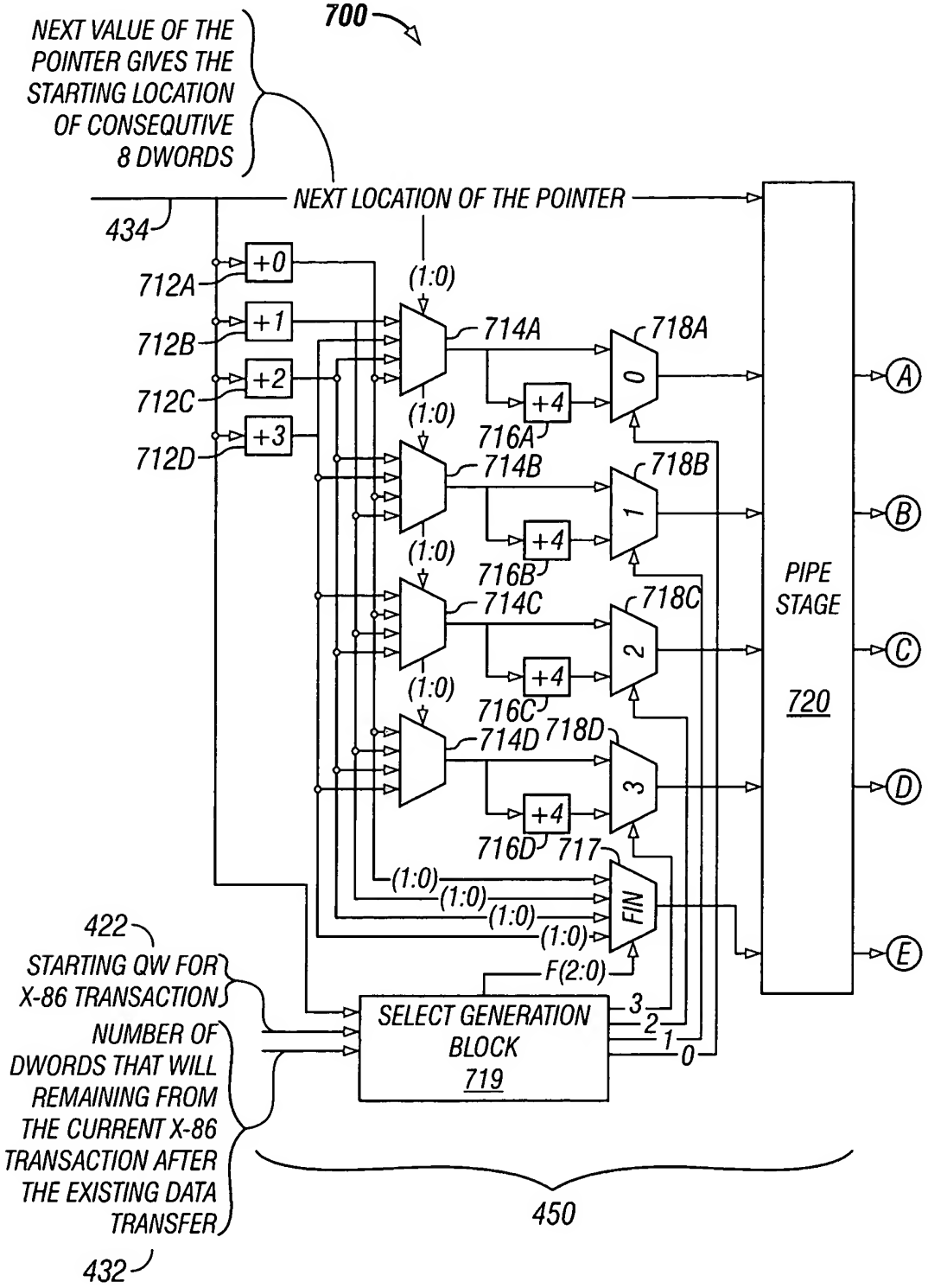
700
NEXT VALUE OF THE POINTER GIVES THE STARTING LOCATION OF CONSEQUTIVE 8 DWORDS
NEXT LOCATION OF THE POINTER
434
712A
712B
712C
712D
+0
+1
+2
+3
(1:0)
714A
714B
714C
714D
+4
716A
716B
716C
716D
718A
718B
718C
718D
0
1
2
3
717
FIN
F(2:0)
PIPE STAGE
720
A
B
C
D
E
422
STARTING QW FOR X-86 TRANSACTION
NUMBER OF DWORDS THAT WILL REMAINING FROM THE CURRENT X-86 TRANSACTION AFTER THE EXISTING DATA TRANSFER
432
SELECT GENERATION BLOCK
719
450

FIG. 7A

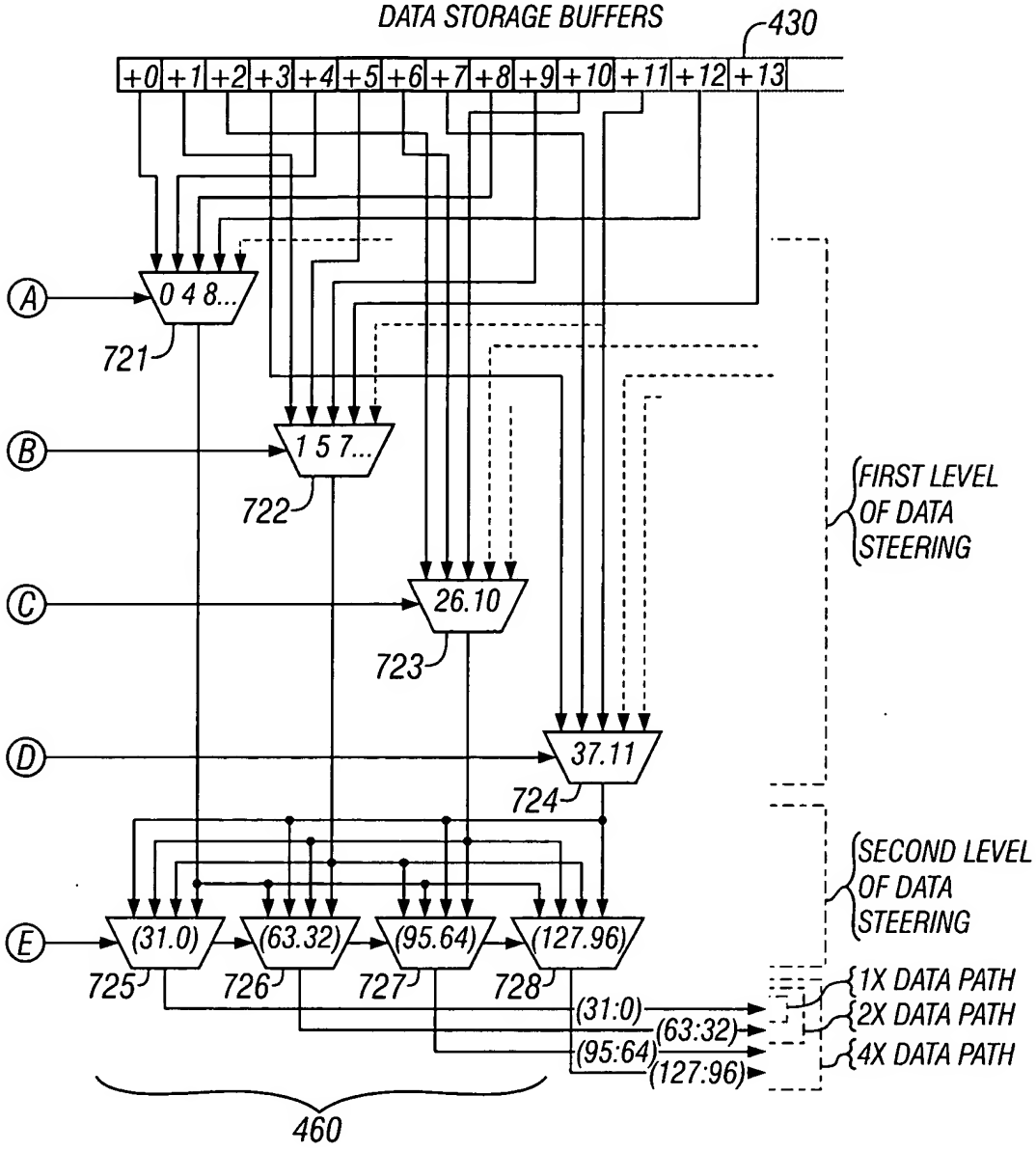
DATA STORAGE BUFFERS
430
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +10 +11 +12 +13
A
0 4 8...
721
B
1 5 7...
722
C
26.10
723
D
37.11
724
FIRST LEVEL OF DATA STEERING
SECOND LEVEL OF DATA STEERING
E
(31.0)
(63.32)
(95.64)
(127.96)
725
726
727
728
(31:0)
(63:32)
(95:64)
(127:96)
1X DATA PATH
2X DATA PATH
4X DATA PATH
460

FIG. 7B

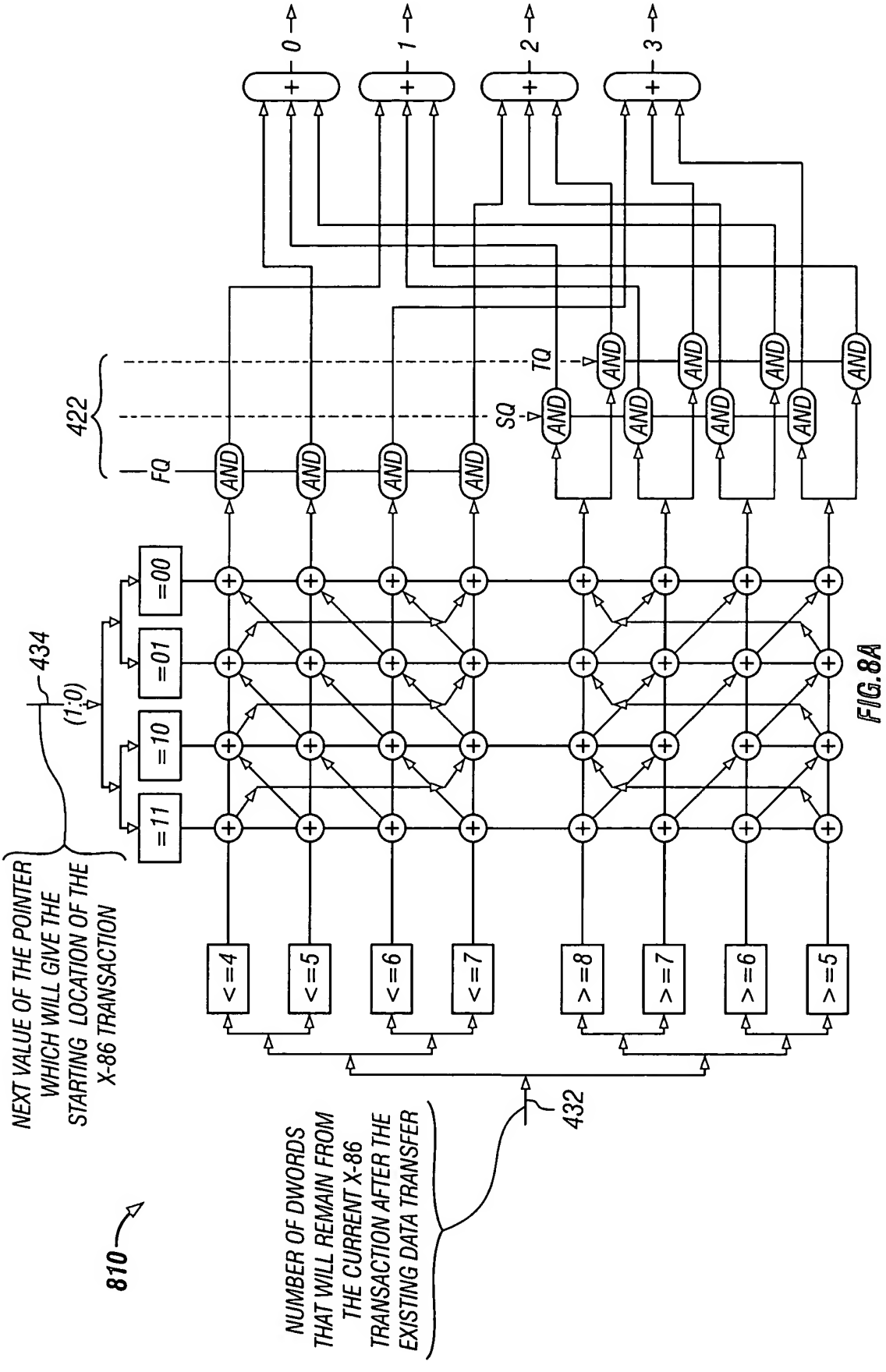
810
NEXT VALUE OF THE POINTER WHICH WILL GIVE THE STARTING LOCATION OF THE X-86 TRANSACTION
434
(1:0)
=00
=01
=10
=11
422
FQ
SQ
TQ
AND
0
1
2
3
+
<=4
<=5
<=6
<=7
>=8
>=7
>=6
>=5
NUMBER OF DWORDS THAT WILL REMAIN FROM THE CURRENT X-86 TRANSACTION AFTER THE EXISTING DATA TRANSFER
432

FIG.8A

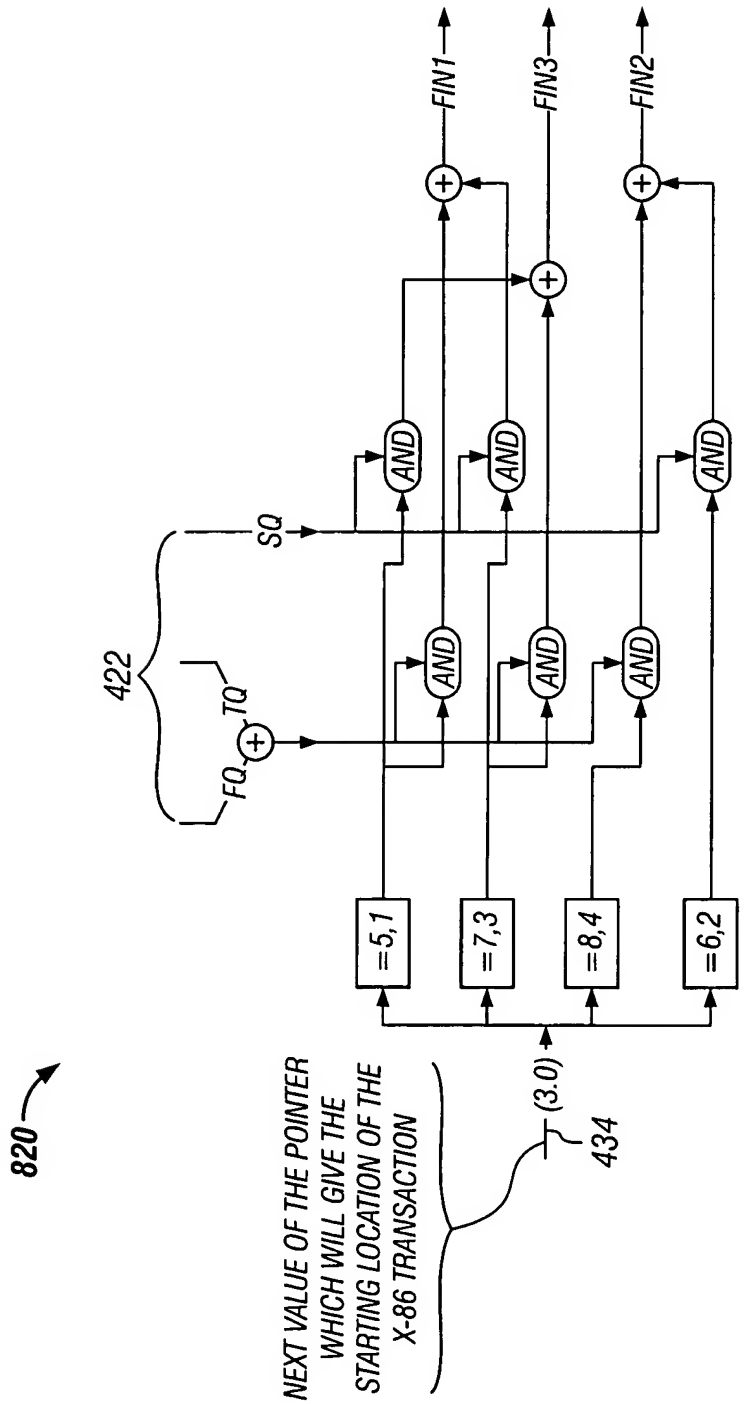
820
NEXT VALUE OF THE POINTER WHICH WILL GIVE THE STARTING LOCATION OF THE X-86 TRANSACTION
(3.0)
434
=5,1
=7,3
=8,4
=6,2
AND
422
FQ
TQ
SQ
FIN1
FIN3
FIN2

FIG.8B

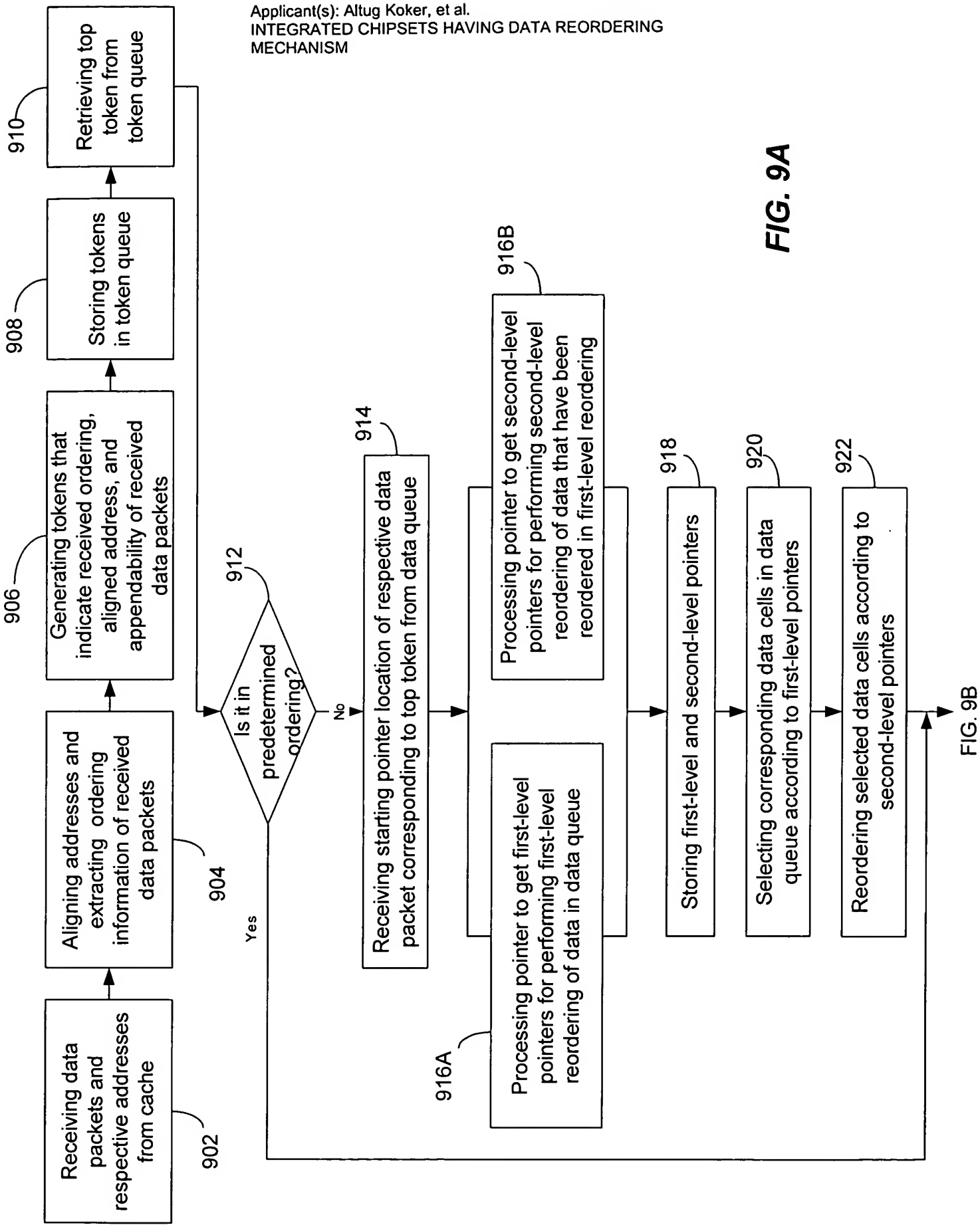
Receiving data packets and respective addresses from cache
902
Aligning addresses and extracting ordering information of received data packets
904
Generating tokens that indicate received ordering, aligned address, and appendability of received data packets
906
Storing tokens in token queue
908
Retrieving top token from token queue
910
Is it in predetermined ordering?
912
Yes
No
Receiving starting pointer location of respective data packet corresponding to top token from data queue
914
Processing pointer to get first-level pointers for performing first-level reordering of data in data queue
916A
Processing pointer to get second-level pointers for performing second-level reordering of data that have been reordered in first-level reordering
916B
Storing first-level and second-level pointers
918
Selecting corresponding data cells in data queue according to first-level pointers
920
Reordering selected data cells according to second-level pointers
922

FIG. 9A

FIG. 9B

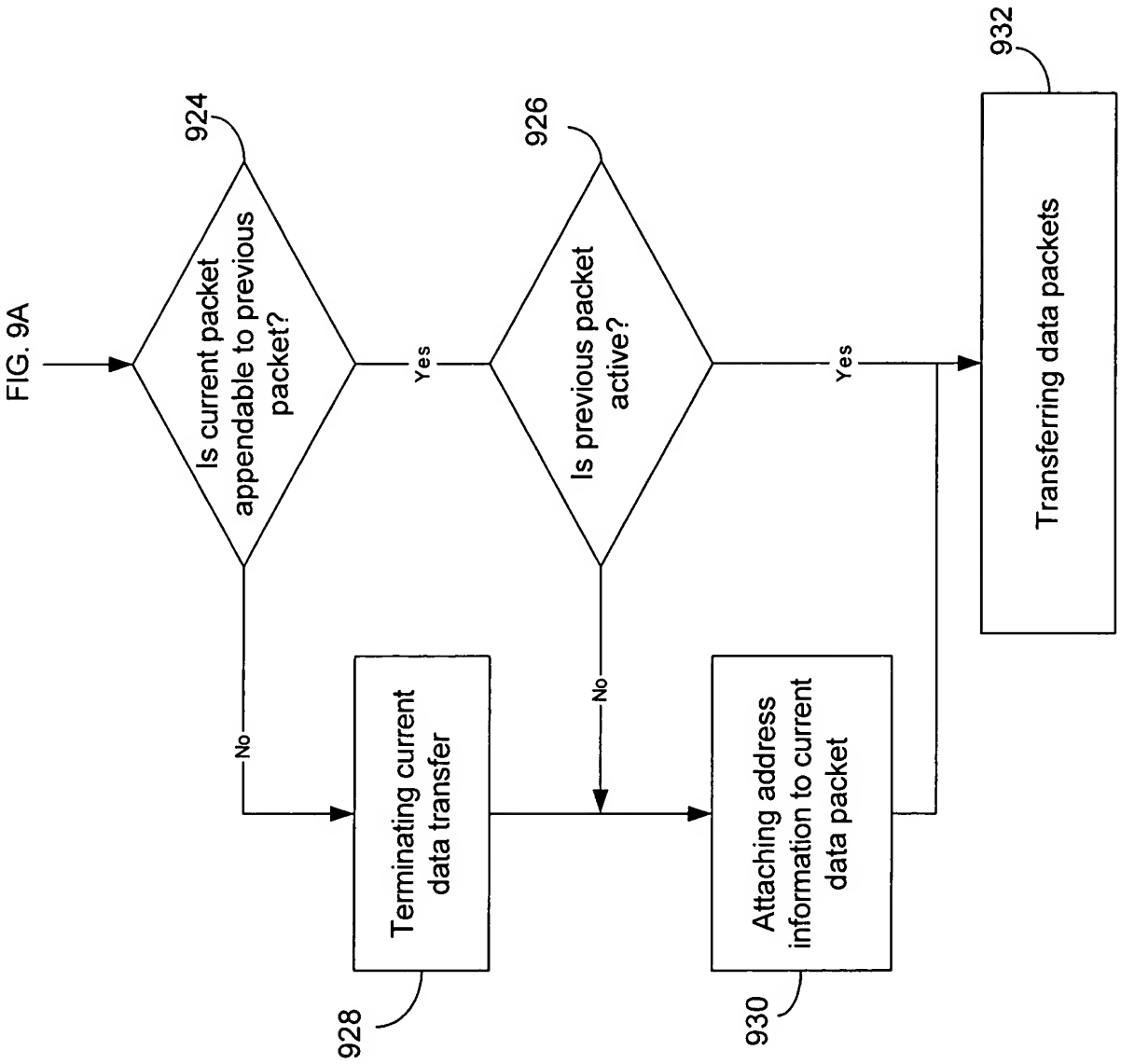
FIG. 9A
924
Is current packet appendable to previous packet?
Yes
926
Is previous packet active?
Yes
932
Transferring data packets
No
928
Terminating current data transfer
No
930
Attaching address information to current data packet

FIG. 9B